

Remarks and Arguments

Claims 1-57 have been presented for examination. Claims 1, 11, 14, 15, 25, 28, 29, 39, 42, 43, 53, 56 and 57 have been amended.

The examiner has required the references to other applications found in the Cross-Reference section and pages 11 and 12 to be updated with any additional available details. In response, the specification has been amended at pages 1, 11 and 12 to enter all available cross reference information.

The drawings have been objected to under 37 C.F.R. §1.83(a) because they do not illustrate the “global status word” recited in the claims. In response, the term “global status word” has been eliminated from the claims.

A new information disclosure statement is filed herewith in order to cite the references that the examiner noted were not properly cited.

Claims 1, 15, 29, 43 and 57 were objected to for an informality involving the omission of a comma. The comma has been added as suggested by the examiner.

Claims 1, 15, 29, 43 and 57 have been rejected under 35 U.S.C. §112, second paragraph, for reciting the phrase “so operates the threads...” which the examiner considers indefinite because it is not clear which entity is operating the threads. In response, claims 1, 15, 29, 43 and 57 have been amended to clearly recite a mechanism or means that operates the threads. For example, claim 1 now recites “a mechanism that operates the threads in a manner that each thread...” Claims 29, 43 and 57 have been amended in a similar manner. Claim 15 is a method claim and, thus does not recite an entity that performs the method steps.

Claims 1, 15, 29, 43 and 57 have been rejected under 35 U.S.C. §112, second paragraph, because the examiner considers the second recitation of the “activity-indicating value” indefinite because it is not clear whether this second recitation refers to the previous recitation of the “activity-indicating value” or whether a new value is being recited. In response, claims 1, 15, 29, 43 and 57 have been amended to remove the recitation of activity-indicating, and inactivity-indicating, values and to recite that the status-word field contains a “value” that indicates whether the associated thread is active or inactive. For example, claim 1 now recites that the computer system comprises a mechanism that operates the threads in a manner that each thread

executes a task-finding routine ... with its associated status-word field containing a value indicating it is active... (lines 6-11). In lines 13-15, claim 1 recites that each thread “sets the contents of its associated status-word field to a value indicating it is inactive...” Claim 1 further recites, in lines 16-20, that each thread, “while the status-word field associated with any other thread contains a value indicating that the other thread is active, searches for a task, and, if it finds one, sets its associated status-word field contents to a value indicating that it is active before attempting to execute a task...” Finally, claim 1 recites in lines 21-24, “if none of the status-word fields associated with other threads contains a value indicating that an associated thread is active...” Consequently, it is believed that claim 1, as amended, clearly sets forth the interaction of the thread with the values in the status-word field. Similar amendments have been made in claims 15, 29, 43 and 57. Therefore, these latter claims also distinctly claim and particularly point out the invention as required by 35 U.S.C. §112, second paragraph.

Claims 1, 15, 29, 43 and 57 have also been rejected under 35 U.S.C. §112, second paragraph, for omitting necessary structural elements between the recited “activity-indicating” values and the recited “inactivity-indicating” values and for omitting necessary structural elements between the recited “global status word” and the recited “separate status-word fields”. In response, claims 1, 15, 29, 43 and 57 have been amended to remove the recitation of the “activity-indicating” values and the “inactivity-indicating” values. Instead, as discussed above, these claims now clearly recite the interaction of the values in the status-word fields and the activity or inactivity of the associated threads. Similarly, the “global status word” has been eliminated from the claims. Consequently, the rejections of claims 1, 15, 29, 43 and 57 for omitting necessary structural elements is hereby traversed.

Claims 1-57 have been rejected under 35 U.S.C. §103(a) as obvious over U.S. Patent No. 6,434,590 B1 (Blelloch) in view of U.S. Published Application No. 2001/0025295 A1 (Kawachiya.) The examiner comments that Blelloch discloses all of the claimed elements with the exception that Blelloch does not explicitly disclose a global status word that is associated with each status-word field. However, the examiner asserts that the Kawachiya reference teaches the use of global reference

objects with states that are substituted for other objects. Consequently, the examiner considers that it would have been obvious to combine Blelloch and Kawachiya by using global status words in Blelloch to signal other threads that the state of an object should be changed.

The present invention relates to a common problem in which a collection of threads that are performing an operation by processing a set of tasks in parallel must determine when the operation is finished. This problem is particularly difficult in situations where the processing of a task can generate other related tasks. Such a condition typically arises in garbage collectors in which objects that contain references to other objects are processed and the other objects must also be processed. In order to avoid inefficiencies, when a thread has completed processing a task and any related tasks, it is desirable for that thread to assist other threads that are still processing tasks by effectively assuming responsibility for processing some of the related tasks of these other threads.

In the present invention, the decision whether the operation is completed is made cooperatively by the threads that are performing the processing. Coordination is achieved by using a set of storage areas (status-word fields) that are accessible to all of the threads. During the processing, each thread can use these fields to not only mark whether it is active or inactive, but also to check whether there are any active threads or whether all threads are inactive, thereby indicating that the operation is finished. If the thread determines that other threads are still processing tasks, it can then search for these related tasks and begin processing some of the related tasks. It should be noted that the completion decision is made cooperatively by the threads and threads directly contact each other to shift processing responsibilities. No central authority that has control over all of the work is involved.

In contrast, the Blelloch system uses an assignment manager as a central authority to assign work to a set of parallel processing elements. Thus, as disclosed, each processing element is assigned a task or set of tasks by the assignment manager (Blelloch, column 2, lines 43-51). During the processing of these tasks other related tasks can arise. The processing element processes the assigned tasks and the related tasks until all tasks and related tasks are finished or until some other condition, such as

a time limit, is reached (Blelloch, column 3, lines 1-8). The processing element then informs the assignment manager either that all tasks have been processed or that additional tasks remain. The processing element and the assignment manager interact by means of a mechanism called a router (Blelloch, column 3, lines 26-37) or a task status buffer (Blelloch, column 3, lines 34-37). Once the processing element has so informed the assignment manager, it then asks the assignment manager whether the operation is complete (Blelloch, column 5, lines 6-10). If the parallel processing operation is not finished, the processing element can then request new assignments from the assignment manager (Blelloch, column 4, lines 60-64). The assignment manager determines whether additional tasks remain to be assigned and if no additional tasks remain to be assigned, the assignment manager declares the operation finished. Therefore, it is clear that in the Blelloch system, a central authority, the assignment manager, distributes tasks and determines whether the operation is finished. The individual processing elements interact with the assignment manager, but do not directly interact. Centralized processing termination techniques have well-known problems. For example, in many systems, the central authority becomes overloaded and slows the entire system down. The Blelloch reference suggests that the assignment manager functions could also be carried out in parallel perhaps on the same processing elements which are performing the tasks. See, for example, Blelloch, column 5, lines 11-19. However, running the assignment manager in parallel only shifts the problem because then some mechanism must be used to determine when the assignment manager functions have terminated. Blelloch neither discloses nor suggests a solution for this latter problem.

The Kawachiya reference discloses a technique for reducing the time required to access objects in memory, where the objects may or may not be accessed by more than one thread. In particular, each object includes a "locality flag" in its header. This flag can be examined by a thread that is attempting to access the object. If the flag indicates that the object can only be accessed by that thread, the thread can skip the normal object locking process, thereby reducing the time required to access the object. To the extent that the locality flag is "global" in the sense that it can be read by any thread, it could be substituted for a task status buffer as disclosed in Blelloch. However,

combining Kawachiya with Blelloch cannot convert Blelloch's centralized assignment manager into a distributed system as disclosed and claimed in the present application. For example, claim 1 now recites, in lines 16-20, that "while the status-word field associated with any other thread contains a value indicating that the other thread is active", each thread "searches for a task, and, if it finds one, sets its associated status-word field contents to a value indicating that it is active before attempting to execute a task..." In Blelloch, the centralized assignment manager reviews the task status buffers and assigns work rather than the threads interacting directly as recited in claim 1. As discussed above combining Kawachiya with Blelloch would not change this basic operation. Further, claim 1 recites, in lines 21-24 "if none of the status-word fields associated with other threads contains a value indicating that an associated thread is active", each thread "terminates its performance of the parallel-execution operation" Thus, each thread decides when the parallel processing operation has terminated. In Blelloch, as modified by Kawachiya, a centralized assignment manager decides when the parallel processing operation is complete. Consequently, claim 1 patentably distinguishes over the cited combination of Blelloch and Kawachiya.

Claims 2-5, 7-9 and 11-14 depend, either directly or indirectly on claim 1 and incorporate the limitations thereof. Consequently, they distinguish over the cited references in the same manner as claim 1. These claims also recite additional elements and limitations not disclosed in Blelloch as modified by Kawachiya. For example, claims 3, 5 and 7-9 recite that the task-finding routine first searches in a work queue associated with its executing thread, but if it does not find any tasks it searches in work queues associated with other threads. In Blelloch/Kawachiya, the assignment manager assigns all tasks. Consequently these claims also distinguish over the cited reference combination for this additional reason.

Claim 15 contains limitation that parallel those in claim 1. Consequently, this claim distinguishes over the cited references in the same manner as claim 1. Claims 16-19, 21-23 and 25-28 depend, either directly or indirectly on claim 15 and incorporate the limitations thereof. Consequently, they distinguish over the cited references in the same manner as claim 15. In addition, these claims contain limitations that parallel

those in claims 2-5, 7-9 and 11-14 and consequently, they distinguish over the cited references in the same manner as those latter claims.

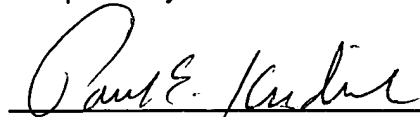
Claim 29 contains limitation that parallel those in claim 1. Consequently, this claim distinguishes over the cited references in the same manner as claim 1. Claims 30-33, 35-37 and 39-42 depend, either directly or indirectly on claim 29 and incorporate the limitations thereof. Consequently, they distinguish over the cited references in the same manner as claim 29. In addition, these claims contain limitations that parallel those in claims 2-5, 7-9 and 11-14 and consequently, they distinguish over the cited references in the same manner as those latter claims.

Claim 43 contains limitation that parallel those in claim 1. Consequently, this claim distinguishes over the cited references in the same manner as claim 1. Claims 44-47, 49-51 and 53-56 depend, either directly or indirectly on claim 43 and incorporate the limitations thereof. Consequently, they distinguish over the cited references in the same manner as claim 43. In addition, these claims contain limitations that parallel those in claims 2-5, 7-9 and 11-14 and consequently, they distinguish over the cited references in the same manner as those latter claims.

Claim 57 contains limitation that parallel those in claim 1. Consequently, this claim distinguishes over the cited references in the same manner as claim 1.

In light of the forgoing amendments and remarks, this application is now believed in condition for allowance and a notice of allowance is earnestly solicited. If the examiner has any further questions regarding this amendment, he is invited to call applicants' attorney at the number listed below. The examiner is hereby authorized to charge any fees or direct any payment under 37 C.F.R. §§1.17, 1.16 to Deposit Account number 02-3038.

Respectfully submitted



Date: 12/22/04

Paul E. Kudirka, Esq. Reg. No. 26,931

KUDIRKA & JOBSE, LLP

Customer Number 021127

Tel: (617) 367-4600 Fax: (617) 367-4656